

**课 程 实 验 报 告**

**课程名称： 大数据分析**

**专业班级： 校交1801**

**学 号： U201814132**

**姓 名： 张浚哲**

**指导教师： 王蔚**

**报告日期： 2020年12月30日**

**计算机科学与技术学院**

**目录**

[实验四 kmeans算法及其实现 1](#_Toc58793873)

[**4.1实验目的** 1](#_Toc58793874)

[**4.2 实验内容** 1](#_Toc58793875)

[**4.3 实验过程** 2](#_Toc58793876)

[4.3.1 编程思路 2](#_Toc58793877)

[4.3.2 遇到的问题及解决方式 2](#_Toc58793878)

[4.3.3 实验测试与结果分析 2](#_Toc58793879)

[**4.4 实验总结** 2](#_Toc58793880)

# 实验四 kmeans算法及其实现

## **4.1实验目的**

1、加深对聚类算法的理解,进一步认识聚类算法的实现；

2、分析kmeans流程,探究聚类算法院里；

3、掌握kmeans算法核心要点；

4、将kmeans算法运用于实际，并掌握其度量好坏方式。

## **4.2 实验内容**

提供葡萄酒识别数据集，数据集已经被归一化。同学可以思考数据集为什么被归一化，如果没有被归一化，实验结果是怎么样的，以及为什么这样。

同时葡萄酒数据集中已经按照类别给出了1、2、3种葡萄酒数据，在cvs文件中的第一列标注了出来，大家可以将聚类好的数据与标的数据做对比。

编写kmeans算法，算法的输入是葡萄酒数据集，葡萄酒数据集一共13维数据，代表着葡萄酒的13维特征，请在欧式距离下对葡萄酒的所有数据进行聚类，聚类的数量K值为3。

在本次实验中，最终评价kmean算法的精准度有两种，第一是葡萄酒数据集已经给出的三个聚类，和自己运行的三个聚类做准确度判断。第二个是计算所有数据点到各自质心距离的平方和。请各位同学在实验中计算出这两个值。

实验进阶部分：在聚类之后，任选两个维度，以三种不同的颜色对自己聚类的结果进行标注，最终以二维平面中点图的形式来展示三个质心和所有的样本点。效果展示图可如图1.1所示。



图1.1 葡萄酒数据集在黄酮和总酚维度下聚类图像（SSE为距离平方和，Acc为准确率）

## **4.3 实验过程**

### 4.3.1 编程思路

本次任务的编程可以分成：

1. 结点数据的加载和管理
2. kmeans初始点的选择
3. 每个结点的预测和对应的sse距离平方和计算
4. k个聚类中心点的生成和迭代收敛
5. 可视化分析。
6. 准确度的计算
7. 结点数据的加载和管理：

本次结点的数据使用的一个结点类（class node）来实现，每一个结点的初始化需要传入对应结点的14个信息，其中第一个信息是其正确的分类，后面的13个信息是其输入的特征。结点记录其真实分类的类别（self.true\_label）和对应的输入（self.input），结点的预测类别初始化为0（self.pre\_label）.

1. k个初始点的选择

函数名：random\_init\_3

入口参数：data（传入的结点集），k（初始的k个点）。

返回值：k个聚类中心的列表。

功能：产生k个初始的聚类中心（此处默认为3）。

实现：首先在所有的结点中随机选择一个点，将其第一维的数据变成0之后对应的是第一个聚类中心点的初始值，然后遍历结点集中的其他的点，以其后13个数据计算欧式距离，选择距离此结点欧式距离最远的结点作为第二个初始结点，然后再遍历剩下的结点，分别算出其与前面两个点的距离，取其距离之和最大的点作为第三个初始点，之后的点类似。次数实现的是3个聚类中心点的初始化选择。

1. 每个结点的预测和对应的sse距离平方和计算

函数名：predict

入口参数：self，k\_classify（当前的k个聚类中心）

返回值：此结点的类别并返回到该类聚类中心的欧式距离。

功能：根据当前的聚类中心，预测此结点的类别并返回到该类聚类中心的欧式距离。

实现：传入k个聚类中心的坐标，然后计算此结点到这k个聚类中心的距离，选择距离最短的一个聚类中心作为其预测的标签，距离使用的是欧式距离，并将此结点和对应预测的聚类中心的欧式距离保存下来。

1. k个聚类中心点的生成和迭代收敛

函数名：classify\_node\_generate

入口参数：all\_node（所有结点组成的结点列表），k（聚类中心的个数）

返回值：classify\_node（计算好的当前的聚类中心）

功能：通过当前预测好的结点生成新的聚类中心。

实现：首先初始化k个全零的14维结点，然后将其第一维的数据分别初始化为0,1,2……k-1，在调用每个结点的预测函数后，得到每个结点的预测类别，根据预测的类别进行聚类中心的定位，每次将对应类别的结点的13维的输入数据加入对应的聚类中心，并记录对应类别的此类分类个数，最后将总数值除以此类分类个数，就能得到对应的新的聚类中心。

迭代使用的是外部循环：

调用每个结点的预测函数之后，将所有结点到其预测的最近的聚类中心的距离加起来，就是整个结点集到其对应的预测中心的距离之和。每次迭代比较新的sse和上一次的sse的差别，当差别小于10^-9时停止收敛。

1. 可视化分析

函数名：druw

入口参数：all\_node（当前所有结点的结点集），accur（计算出来预测的准确率），loss（计算出来的总体sse）

返回值：无。

功能：通过当前结点集合预测的情况，画图其对应的聚类图，并显示分类的准确率和总体sse

实现：首先使用PCA降维将13维数据降低到2维，然后将三类的结点使用不同的颜色进行画图即可。

1. 准确度的计算

函数名：true\_classify\_node\_generate

入口参数：all\_node（所有结点组成的结点列表），k（聚类中心个数）

返回值：classify\_node（真实的聚类中心）

功能：使用结点的真实标签作为聚类中心产生的依据，产生真实的聚类中心。

实现：每次遍历结点集中的结点，使用结点本身真实的标签进行计算，将对应结点的数据加在聚类中心之上，并统计每个聚类中心的数目，最后进行平均，便得到了所有聚类中心的坐标。

函数名：get\_label\_classify\_node

入口参数：pre\_classify\_node（传入的需要获得真实标签的结点）, true\_nodes（真实的聚类中心）

返回值：pre\_classify\_node（返回获得真实标签的结点）

功能：传入一个结点和真实的聚类中心，获得该结点对应的真实标签。

实现：传入一个结点，遍历所有的真实聚类中心，分别计算其与所有聚类中心的距离，取距离该结点最近的聚类中心作为该结点的真实分类。

### 4.3.2 遇到的问题及解决方式

1.画图

数据本身的维度是13维度的，但是在图中只能展现两个维度，如果只是随机的选取其中的两个维度，可能会丢失数据中其他很多的信息，因此这里使用pca降维将原始数据从13维度降低到2维，从而能够在图上画出对应的统计图而又不损失原始数据的很多信息。

2.真实标签的确定

按照之前的编程思路，是可以完成结点的分类和聚类中心的迭代的，但是此时结点分类由于kmeans是无监督学习，所以此时得到的分类标签和真实的标签不一定对的上号，因此需要将最后收敛的聚类中心和真实的聚类中心进行距离的比较，每一个生成的聚类中心都取离其最近的聚类中心的标签作为其标签，这样就可以获得每个聚类的真实类别，就可以获得算法的准确率。

3.聚类中心的产生和迭代

为了方便程序实现，本次采用的是结点集，结点集是所有结点组成的列表，其中每个结点拥有对应的13维数据，以及对应的预测标签和真实标签，每个结点拥有一个预测函数，不仅能根据当前离此结点最近的聚类中心进行分类，但能将对应的分类的sse（距离平方和）进行计算，而聚类中心又可以通过结点的预测标签进行产生，这样就解决了聚类中心和产生和迭代的问题了。

### 4.3.3 实验测试与结果分析

我们加载数据集，分析其数据的组成方式，打印出其所有数据的前3项，并打印出每一个数据的维数以及所有数据的个数，如图1.1所示：

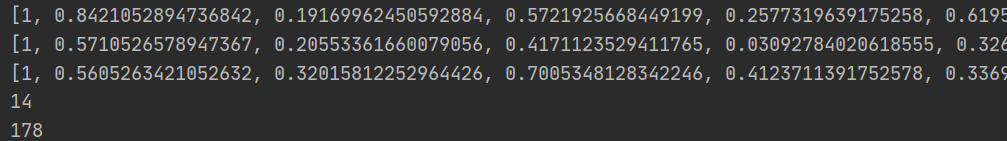


图1.1

上图中，前三行是数据集中的前三个数据。14是每个数据的维度（包括标签），178是所有结点的个数。

然后进行初始聚类中心的初始化，并且将其标签分别初始为0,1，2（为了方便后面的聚类中心的下标访问），如图1.2所示：

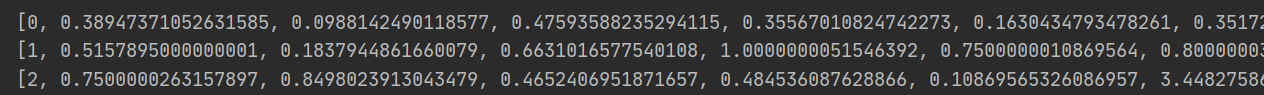


图1.2

然后进行聚类中心的迭代，迭代过程如图1.3所示：

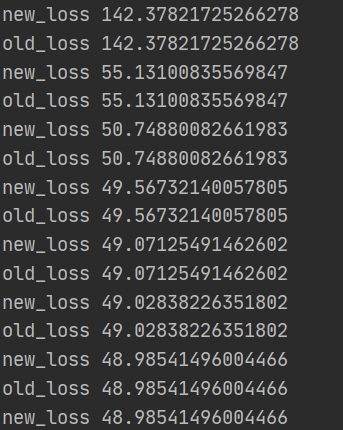


图1.3

这里的old\_loss和new\_loss分别是此次迭代之前和此次迭代之后的sse（所有结点到其预测聚类中心的距离平方和）。

迭代了8次收敛。最后的sse总和是48.9854.

最后生成的聚类分析图如图1.4所示：

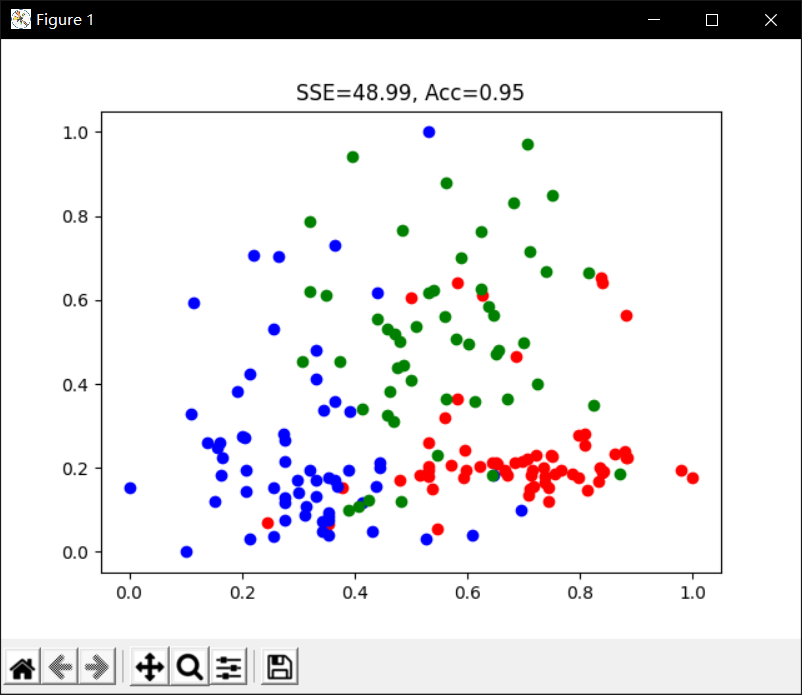


图1.4

此处的图可以反映原始数据大概被分成了了三类，sse为48.99，Acc（准确率）为0.95，此处的图和任务书中的图不同主要是因为任务书中使用的是总酚和黄酮两个标签，由于不知道对应csv中那个标签是总酚和黄酮，因此此处直接使用的是所有13维数据pca降维到2维产生的图。

## **4.4 实验总结**

kmeans实验虽然本身实验流程比较长，但是由于之前有过一点了解，因此此次实验总体来说花的时间不长，但是对这个算法有了解和复现这个算法还是有很大的不同的，首先是对数据的管理，其次是蕨类中心的生成，而且此次实验还触及到之前一个没有接触的问题，就是如何给评价无监督算法的效果。本次实验中使用的方法是将生成的聚类中心和真实的聚类中心进行比较，将用离生成的聚类中心最近的真实聚类中心的标签作为生成聚类中心的标签,但是这样做的前提是算法本身具有一定的准确性，当算法本身的效果不好，或者k较多，可能效果就不是很好，特别是当两个真实的聚类中心相隔很近的时候在获取真实标签的时候就很容易获取错。

kmeans是很经典的一个无监督聚类算法，本身的效果就还不错，通过此次的实验，使我加深了对kmeans的理解，了解了kmeans中的一些细节，也遇到了一些之前没有出现过的问题，比如无监督学习的评测等问题。总的来说，在算法和编码能力上都有一定的提高，并且加强了无监督学习这方面的理解。